



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/767,365	01/22/2001	Sheng Liang	2006579-0558 (CTX-199)	2538
69665 7590 11/05/2008 CHOATE, HALL & STEWART / CITRIX SYSTEMS, INC. TWO INTERNATIONAL PLACE BOSTON, MA 02110				
EXAMINER				
TRAN, QUOC A				
ART UNIT		PAPER NUMBER		
2176				
MAIL DATE		DELIVERY MODE		
11/05/2008		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

# Office Action Summary

## Application No.

09/767,365

## Applicant(s)

LIANG ET AL.

## Examiner

Quoc A. Tran

## Art Unit

2176

**Period for Reply** -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 08 September 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1, 4-6, 8-13 and 15 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1, 4-6, 8-13 and 15 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 22 May 2008 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/C)
- 4) ☐ Interview Summary (PTO-413)
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_
- Paper No(s)/Mail Date \_\_\_\_\_

### DETAILED ACTION

This is a **Non-Final** Office Action in response to petition to revive application, which is granted on 09/09/2008 further in view of Amendment/Remarks filed 05/22/2008. Claims 1, 4-6, 8-13 and 15 are pending. Claims 1, 6, 8-11, 13 and 15 are independent claims. Applicants have amended claims 1, 4-5, 8, 10-13, and 15, which claimed priority of which claimed benefit of 60/228,904 filed **08-29-2000** to assignee ["Stratum8"].

### *Specification*

The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. Claims 1, 4-6, 8; 11-13 and 15 recite the limitation "**efficiently**". However, there is not any support for the term "**efficiently**" in the Applicant's disclosure.

In additions, claim 15 recites the limitation "**computer-readable-medium**". However, there is not any support for the term "**computer-readable-medium**" in the Applicant's disclosure. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o).

Correction is required.

***Claims Rejections – 35 U.S.C. 112, Second Paragraph***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

***The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter, which the applicant regards as his invention.***

Claims 1, 4-6, 8; 11-13 and 15 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention, because of the following reason:

Claims 1, 4-6, 8; 11-13 and 15 recite the limitation "***efficiently***" parsing received data file.... In the claims Pages 2-5, render the claims indefinite. Since there is not any support for the term "***efficiently***" in the Applicant's disclosure; thus renders the claims indefinite, since one of ordinary skill in the art would not be reasonably appraised what is being claimed.

In the interest of compact prosecution, the application is further examined against the prior art, as stated below, upon the assumption that the applicants may overcome the above stated rejections under 35 U.S.C. 112.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

*(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.*

**Claims 1, 4-6, 8-12 and 15** are rejected under 35 U.S.C. 103(a) as being unpatentable by Emmelmann US 20030074634A1 filed 11/24/1999 [hereinafter "Emmelmann"], in view of Krishnamurthy et al. US 20060242145A1 Provisional No. 60/226,479 filed 08/18/2000 [hereinafter "Krishnamurthy"].

*Regarding independent claim 1,*

Emmelmann teaches:

**a computer-implemented method for efficiently parsing received data file, comprising: receiving a data file;**

(See Paragraph [0116])→ Emmelmann discloses this limitation that is reads and parses the requested component page resulting in an abstract syntax tree (AST).)

**retrieving a stored version of the data file and a syntax tree comprising nodes and tokens representing data within the data file, the tree include at least one static node.**

(See Fig. 7 and Fig. 10 and Paragraph [0022])→ Emmelmann discloses a mechanism to

generate pages from such page templates that allow multiple independent and interactive components per dynamic page, and a browser-based editor to place and modify components on page templates. Also Emmelmann further discloses reads and parses the requested component page resulting in an abstract syntax tree (AST), [See Para [0116]. Also Emmelmann further discloses the AST contains two kinds of nodes, one for browser code and the other one for components marked on the page. Each node representing a component with content in turn contains a cb-list that represents the content of the component. In terms of a tree, the cb-lists connect all the children of a node or of the tree root [para 0123]. Also see Emmelmann at Para [0320→347], discloses the abstract syntax tree (AST) includes static and dynamics nodes wherein the syntax of a tag definition in EBNF. Terminal tokens are printed in bold letters while Nonterminals are written in italics. This allows heitml [an HTML web site consists of HTML pages, a heitml web site consists of heitml pages. heitml is an extension of HTML, so heitml pages can contain normal HTML code.

This interpretation is supported by the applicant's disclosure, which is stated, "syntax tree," and "nodes and tokens," is a template/token tree wherein an AST (abstract syntax tree) modified to contain "nodes and tokens". Each node of the tree is initially label as a static node..." (See current disclosure page 11 lines 1-10). In its broadest reasonable interpretation, the web page (HTML page) parses and a template/token tree is being built, which contains "nodes and tokens," For example, a template provides an initial definition for the Web page. It is used to generate a tree of objects that can be used to generate the Web page. The template and corresponding

object tree can be modified dynamically bases on the associations and custom logic (syntax).

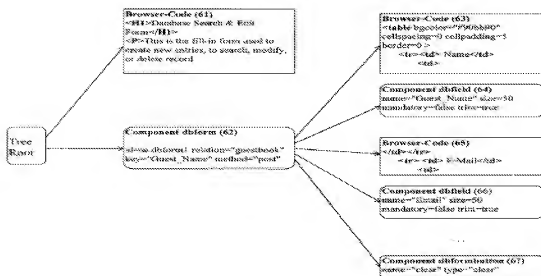
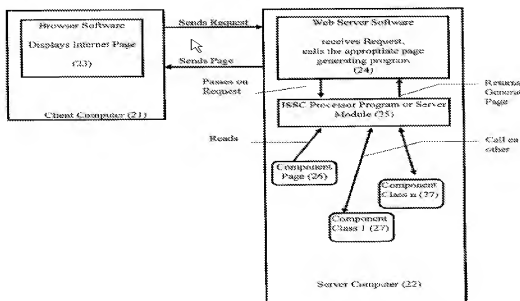


Fig. 10: Example Abstract Syntax Tree (AST)

In addition, Emmelmann does not expressly teach, but Krishnamurthy teaches:

**comparing the stored version of the data file with the received data file to identify non-matching content in the received data file, parsing only the non-matching content of the received data file to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file;**

(See Figure. 5 and page 5 Para [0077] --> Page 6 Para [0083]→ Krishnamurthy discloses this limitation that is defines a different between HTML pages by comparison utilized HTMLDiff algorithm and the HTML as a hierarchical document and compare the parse tree or abstract syntax tree representations of the documents, using subtree equality (or some weaker measure) as a basis for comparison. In this case, a subtree representing nodes and tokens to one of the subtrees and identify only differences of the HTML. In this case the tokens1 is comparing to tokens2 to result the FlatDiff as shows in Fig. 5.)

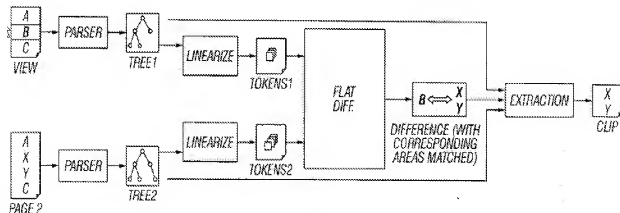


FIG. 5



**replacing at least one static node of the syntax tree with a token; and  
creating a mapping from each token to one of the subtrees.**

(See Figure. 5 and page 5 Para [0077] --> Page 6 Para [0083], and Para [0129]→  
Krisnamurthy discloses the Improving TreeDiff Performance: Subtree matching wherein  
the difference algorithms compute the mapping from the nodes in T1 to the nodes in  
T2. Given such a mapping, we can identify whether the distinguished node(s) from the  
old page is preserved in some form inside the new page, and if so, the subtree rooted at  
that node in T2 is the clip to be extracted.)

This interpretation is supported by the applicant's disclosure, which is stated,  
"System 500 ... newly retrieved version of the page and the cached copy of the page  
are compared using a binary "diff" algorithm, which identifies the differences between  
the binary representation of two documents..." (see current disclosure page 11 lines  
11→Line 25).

It would have been obvious to a person of ordinary skill in the art at the time the  
invention was made to have modified Emmelmann's mechanism to generate pages  
from such page templates that allow multiple independent and interactive components  
per dynamic page, and a browser-based editor to place and modify components on  
page templates wherein the server reads and parses the requested component page  
resulting in an abstract syntax tree (AST) utilized nodes that is representing a  
component with content in turn contains a cb-list that represents the static and dynamic  
page at the server, to include a means of said comparing the stored version of the data

file with the received data file to identify non-matching content in the received data file, parsing only the non-matching content of the received data file to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file and replacing at least one static node of the syntax tree with a token; and creating a mapping from each token to one of the subtrees of Krishnamurthy. because they are from the same field of endeavor of Viewing and parsing HTML using AST (Abstract Syntax tree), and provides a predictable result of said to generate HTML page that is interactive objects can be combined, nested, parameterized, and programmed very flexibly and allows reusing cooperating interactive objects instead of rewriting everything from scratch each time [Emmelmann Para [0322] and further provide the TreeDiff Performance that is defines a different between HTML pages by comparison utilized HTMLDiff algorithm and the HTML as a hierarchical document and compare the parse tree or abstract syntax tree representations of the documents, using subtree equality (or some weaker measure) as a basis for comparison HTML page segments to be extracted [See Krishnamurthy at page 5 Para [0077] --> Page 6 Para [0083].)

**Claim 4,**

Emmelmann teaches:

**wherein the data file is a web page,**

(See Fig. 7 and Paragraph [0100-0101]→ Emmelmann discloses web page is data file.)

***Claim 5,***

Emmelmann teaches:

**wherein the data file is an HTML file,**

(See Fig. 7 and Paragraph [0100-0101]→ Emmelmann discloses data file is HTML file.)

***Regarding independent claim 6,***

Claim 6 is fully incorporated similar subject of claims 1 and 4-5 cited above, and are similarly rejected along the same rationale. Thus, Emmelmann and Krishnamurthy disclose every limitation of Claim 6 and provide proper reasons to combine, as indicated in the above rejections for Claims 1 and 4-5.

***Regarding independent claim 10,***

Claim 10 is fully incorporated similar subject of claim 1 cited above, and is similarly rejected along the same rationale. Thus, Emmelmann and Krishnamurthy disclose every limitation of Claim 10 and provide proper reasons to combine, as indicated in the above rejections for Claim 1.

In addition, Emmelmann teaches:

**method of providing derivative services, comprising: receiving a request for derivative services content from a customer; retrieving data from a plurality of primary service providers on behalf of the customer,**

(See Fig. 8 and Paragraph [0115]→ Emmelmann discloses the user invokes a web browser on the client computer and interacts with the user interface of the browser to select a page possibly being a component page (31). The browser sends (32) a request to the web server. The web server receives and analyzes (33) the request, and when a component page is requested, the web server calls the component processor (34). Most web servers available today can be parameterized to do this.)

**identifying static content that has been previously retrieved from the primary service providers and stored, and corresponding template/token trees that have also been stored; identifying dynamic content that differs from the previously retrieved content;**

(See Fig. 7 and Fig. 10 and Paragraph [0022]→ Emmelmann discloses a mechanism to generate pages from such page templates that allow multiple independent and interactive components per dynamic page, and a browser-based editor to place and modify components on page templates. Also Emmelmann further discloses reads and parses the requested component page resulting in an abstract syntax tree (AST), [See Para [0116]. Also Emmelmann further discloses the AST contains two kinds of nodes, one for browser code and the other one for components marked on the page. Each node representing a component with content in turn contains a cb-list that represents the content of the component. In terms of a tree, the cb-lists connect all the children of a node or of the tree root [para 0123]. Also see Emmelmann at Para [0320→347],

discloses the abstract syntax tree (AST) includes static and dynamics nodes wherein the syntax of a tag definition in EBNF. Terminal tokens are printed in bold letters while Nonterminals are written in italics. This allows heitml [an HTML web site consists of HTML pages, a heitml web site consists of heitml pages. heitml is an extension of HTML, so heitml pages can contain normal HTML code.

This interpretation is supported by the applicant's disclosure, which is stated, "syntax tree," and "nodes and tokens," is a template/token tree wherein an AST (abstract syntax tree) modified to contain "nodes and tokens". Each node of the tree is initially label as a static node..." (See current disclosure page 11 lines 1-10). In its broadest reasonable interpretation, the web page (HTML page) parses and a template/token tree is being built, which contains "nodes and tokens," For example, a template provides an initial definition for the Web page. It is used to generate a tree of objects that can be used to generate the Web page. The template and corresponding object tree can be modified dynamically bases on the associations and custom logic (syntax).

In addition, Emmelmann does not expressly teach, but Krishnamurthy teaches:

**parsing the dynamic content to form subtrees; adding tokens  
to the template/token trees; mapping the tokens to the subtrees;  
creating at least one content page comprising the retrieved data; and  
providing the created pages to the customer.**

(See Figure. 5 and page 5 Para [0077] --> Page 6 Para [0083]→ Krisnamurthy  
discloses this limitation that is defines a different between HTML pages by comparison

utilized *HTMLDiff* algorithm and the HTML as a hierarchical document and compare the parse tree or abstract syntax tree representations of the documents, using subtree equality (or some weaker measure) as a basis for comparison. In this case, a subtree representing nodes and tokens to one of the subtrees and identify only differences of the HTML. In this case the tokens1 is comparing to tokens2 to result the FlatDiff as shows in Fig. 5. Also Krisnamurthy further discloses the Static Clips vs. Dynamic Clips, wherein dynamic programming algorithms that are able to recognize repetitions of certain patterns in the input sets, then utilized FlatDiff to Computing Page Difference between Unstructured Documents. A parser is used to first transform a web page into an abstract syntax tree (AST). The tree is then linearized into a sequence of tokens, which consist of markup elements (defined by the markup language syntax and denoting structure, semantics, formatting or other information as discussed earlier) and text strings that represent the content. The token sequences corresponding to the view and the new page are then fed into the FlatDiff stage, which computes a shortest edit sequence using our extensions to a well-known flat edit sequence calculation algorithm. By locating the matching tokens in the original ASTs, the extraction stage outputs the desired clip. This is generally discloses at Para [0012] and [0076-->0128].)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Emmelmann's mechanism to generate pages from such page templates that allow multiple independent and interactive components per dynamic page, and a browser-based editor to place and modify components on

page templates wherein the server reads and parses the requested component page resulting in an abstract syntax tree (AST) utilized nodes that is representing a component with content in turn contains a cb-list that represents the static and dynamic page at the server, to include a means of said parsing the dynamic content to form subtrees; adding tokens to the template/token trees; mapping the tokens to the subtrees; creating at least one content page comprising the retrieved data; and providing the created pages to the customer of Krishnamurthy, because they are from the same field of endeavor of Viewing and parsing HTML using AST (Abstract Syntax tree), and provides a predictable result of said to generate heitml page that is interactive objects can be combined, nested, parameterized, and programmed very flexibly and allows reusing cooperating interactive objects instead of rewriting everything from scratch each time [Emmelmann Para [0322] and further provide the TreeDiff Performance that is defines a different between HTML pages by comparison utilized HTMLDiff algorithm and the HTML as a hierarchical document and compare the parse tree or abstract syntax tree representations of the documents, using subtree equality (or some weaker measure) as a basis for comparison HTML page segments to be extracted [See Krishnamurthy at page 5 Para [0077] --> Page 6 Para [0083].)

*Regarding independent claim 8,*

Claim 8 are fully incorporated similar subject of claims 1, 4-5 and 10 cited above, and are similarly rejected along the same rationale. Thus, Emmelmann

and Krishnamurthy disclose every limitation of Claim 8 and provide proper reasons to combine, as indicated in the above rejections for Claims 1-45 and 10.

*Regarding **independent claim 9,***

Claim 9 is fully incorporated similar subject of claims 1, 4-5 and 10 cited above, and is similarly rejected along the same rationale. Thus, Emmelmann and Krishnamurthy disclose every limitation of Claim 9 and provide proper reasons to combine, as indicated in the above rejections for Claims 1, 4-5 and 10.

*Regarding **independent claim 11,***

Claim 11 is fully incorporated similar subject of claim 1 cited above, and is similarly rejected along the same rationale. Thus, Emmelmann and Krishnamurthy disclose every limitation of Claim 11 and provide proper reasons to combine, as indicated in the above rejections for Claim 1.

In addition, Emmelmann teaches:

**retrieving a second data file, the second data file associated  
with the first data file,**

(See Fig. 7 and Fig. 10 and Paragraph [0022]→ Emmelmann discloses a mechanism to generate pages from such page templates [e.g. first data file, second data file] that allow multiple independent and interactive components per dynamic page, and a browser-based editor to place and modify components on page templates.)



***Claim 12,***

Claim 12 is fully incorporated similar subject of claim 10 cited above, and is similarly rejected along the same rationale. Thus, Emmelmann and Krishnamurthy disclose every limitation of Claim 12 and provide proper reasons to combine, as indicated in the above rejections for Claim 10 [e.g. first data file if data file].

***Regarding independent claim 15,***

Claim 15 recites a computer program product to implement a method recited in Claim 1. Thus, Emmelmann and Krishnamurthy disclose every limitation of Claim 15 and provide proper reasons to combine, as indicated in the above rejections for Claim 1 - Also See Emmelmann at Para [0393], disclose persistent memory.)

**Claim 13** is rejected under 35 U.S.C. 103(a) as being unpatentable by **Emmelmann** US 20030074634A1 filed 11/24/1999 [hereinafter “Emmelmann”], in view of **Krishnamurthy** et al. US 20060242145A1 Provisional No. 60/226,479 filed 08/18/2000 [hereinafter “Krishnamurthy”], further in view of **Geller** US 20040199575A1 filed 11/24/1999 [hereinafter “Emmelmann”],

*Regarding independent claim 13,*

Claim 13 recites a system [e.g. an identification engine, a cache, a comparison engine, a parsing engine, a content server] to implement a method recited in Claim 1. Thus, Emmelmann and Krishnamurthy disclose every limitation of Claim 13 and provide proper reasons to combine, as indicated in the above rejections for Claim 1 - Also See Emmelmann at Para [0086], a GUI system; Para [0188] ISSC Processor; Para [0393], persistent memory; Para [0011] means to compare; Para [0121-0122], a parser; and Para [0079] discloses a content sever.)

In addition, Emmelmann and Krishnamurthy do not expressly teach, but Geller teaches:

**At least one virtual browser for retrieving content from primary content server,**

(See Fig. 2B and Para 0073)→Geller discloses The BB software comprises a virtual browser 205, a virtual server 209, virtual plugins 207 and interactive BB files 211 as illustrated at 203 in FIG. 2A. The embodiment in FIG. 2B contains software for

producing an Interactive Media Site (IMS) on a user's host system, in this case a PC. Also Geller further discloses The VSMS performs the same function as a merchant's remote Internet server in a common E-Commerce System, except the BB is loaded into the memory of the user's local "host device." at Para [0019].)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Emmelmann and Krishnamurthy method to include am means of said related to the uses of at least one virtual browser for retrieving content from primary content server as taught by Geller. One of the ordinary skills in the art would have been motivated to perform such a modification, because they are from the same field of endeavor of updates to any affiliates or merchant pages are available can cause the HTML page generator to create dynamic or static pages as required according to the nature of the internet of WWW needs and provides a predictable result of said to generate heitml page that is interactive objects can be combined, nested, parameterized, and programmed very flexibly and allows reusing cooperating interactive objects instead of rewriting everything from scratch each time [Emmelmann Para [0322] and further provide the TreeDiff Performance that is defines a different between HTML pages by comparison utilized HTMLDiff algorithm and the HTML as a hierarchical document and compare the parse tree or abstract syntax tree representations of the documents, using subtree equality (or some weaker measure) as a basis for comparison HTML page segments to be extracted [See Krishnamurthy at page 5 Para [0077] --> Page 6 Para [0083].)

It is noted that any citations to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. See, MPEP 2123.

### ***Response to Arguments***

Applicant's arguments with respect to claims 1, 4-5, 6, 8-13 and 15 have been considered but are moot in view of the new ground(s) of rejection [See above rejection for details].

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on 9AM - 5PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on 571-272-4137. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

*/Quoc A. Tran/*  
Patent Examiner

*/Doug Hutton/*  
Doug Hutton  
Supervisory Primary Examiner  
Technology Center 2100